SENTI-MD: IMDb Movie Review Sentiment Analysis

Team Members: Prateek P (PES1UG23AM211) & Vivian Philip (PES1UG23AM907)

The Challenge We Tackled

Picture this: you're scrolling through hundreds of movie reviews, trying to figure out if people actually like that new film. Tedious, right?

We built SENTI-MD to solve exactly that problem. Our goal was simple but ambitious: create a machine learning classifier that reads IMDb reviews and instantly tells you whether the sentiment is positive or negative.

The real challenge? Human language is messy, nuanced, and full of sarcasm. Teaching a computer to understand that is no small feat.

Why This Matters

Speed at Scale

Analyse thousands of reviews in seconds, not weeks. What used to take a team days now happens instantly.

Actionable Insights

Turn messy text into clean data. Get a clear picture of audience reception to guide marketing decisions.

Real Business Value

Film studios and streaming platforms can quickly gauge public opinion and adjust their strategies accordingly.

Our Approach: The SENTI-MD Pipeline

Raw Reviews

50,000 IMDb movie reviews, full of HTML tags, emojis, and everything in between

Pre-processing

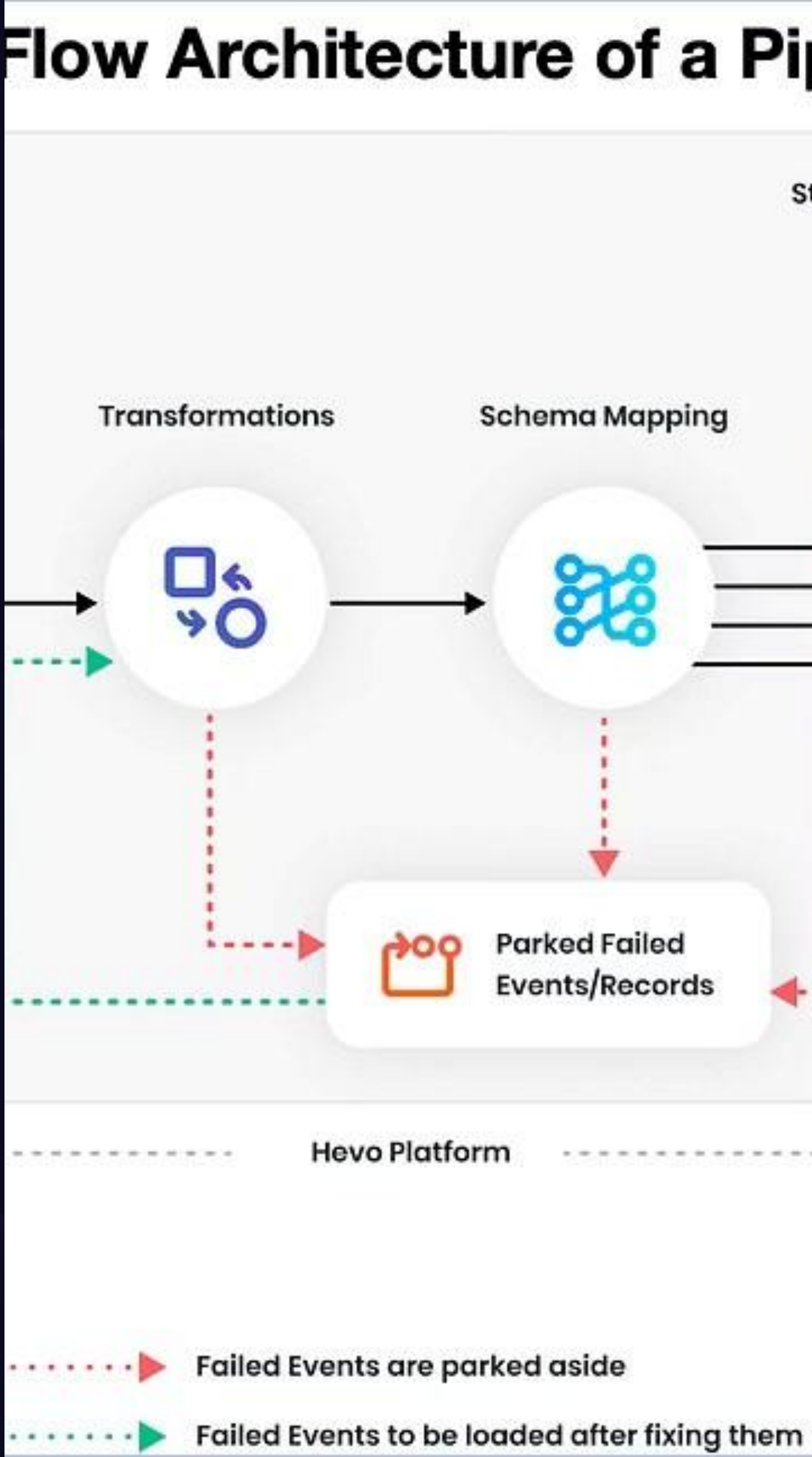
Clean the noise: remove HTML, punctuation, stop words, and normalise the text

Feature Extraction

Convert words into numbers using TF-IDF with 3-grams to capture context

Classification

Train our model to predict: positive or negative?


Flow Architecture of a Pi
St
Transformations
Schema Mapping
Parked Failed Events/Records
Hevo Platform
Failed Events are parked aside
Failed Events to be loaded after fixing them

Getting the Data Ready

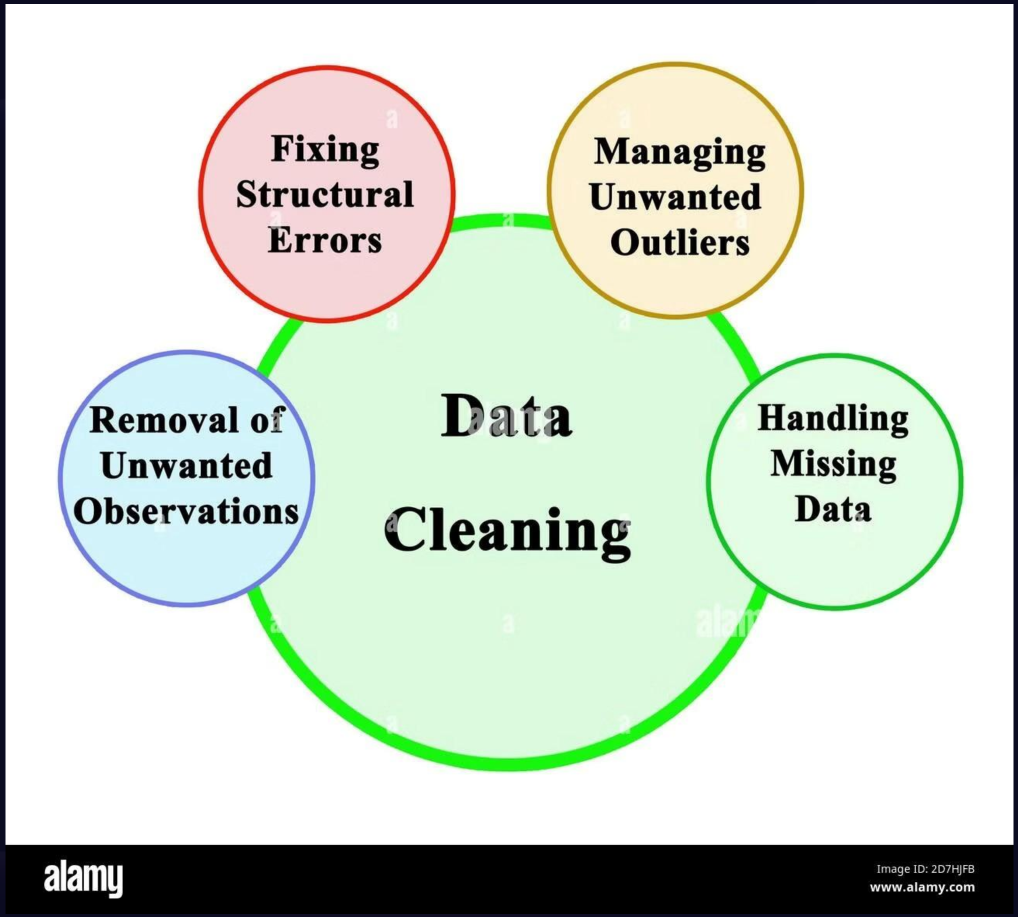
The Dataset

We worked with the classic IMDb Movie Review Dataset containing **50,000 reviews** 3 a proper benchmark in the NLP community. Half positive, half negative, perfectly balanced for training.

Pre-processing Steps

- Stripped out HTML tags and special characters
- Converted everything to lowercase for consistency
- Removed common stop words using NLTK
- Applied tokenisation to break text into meaningful units

This cleaning process transformed messy, real-world text into something our model could actually learn from.



Feature Extraction: The Secret Sauce

Here's where things get interesting. We used **TF-IDF (Term Frequency-Inverse Document Frequency)** with 3-grams to convert our cleaned text into numerical features.

Why TF-IDF?

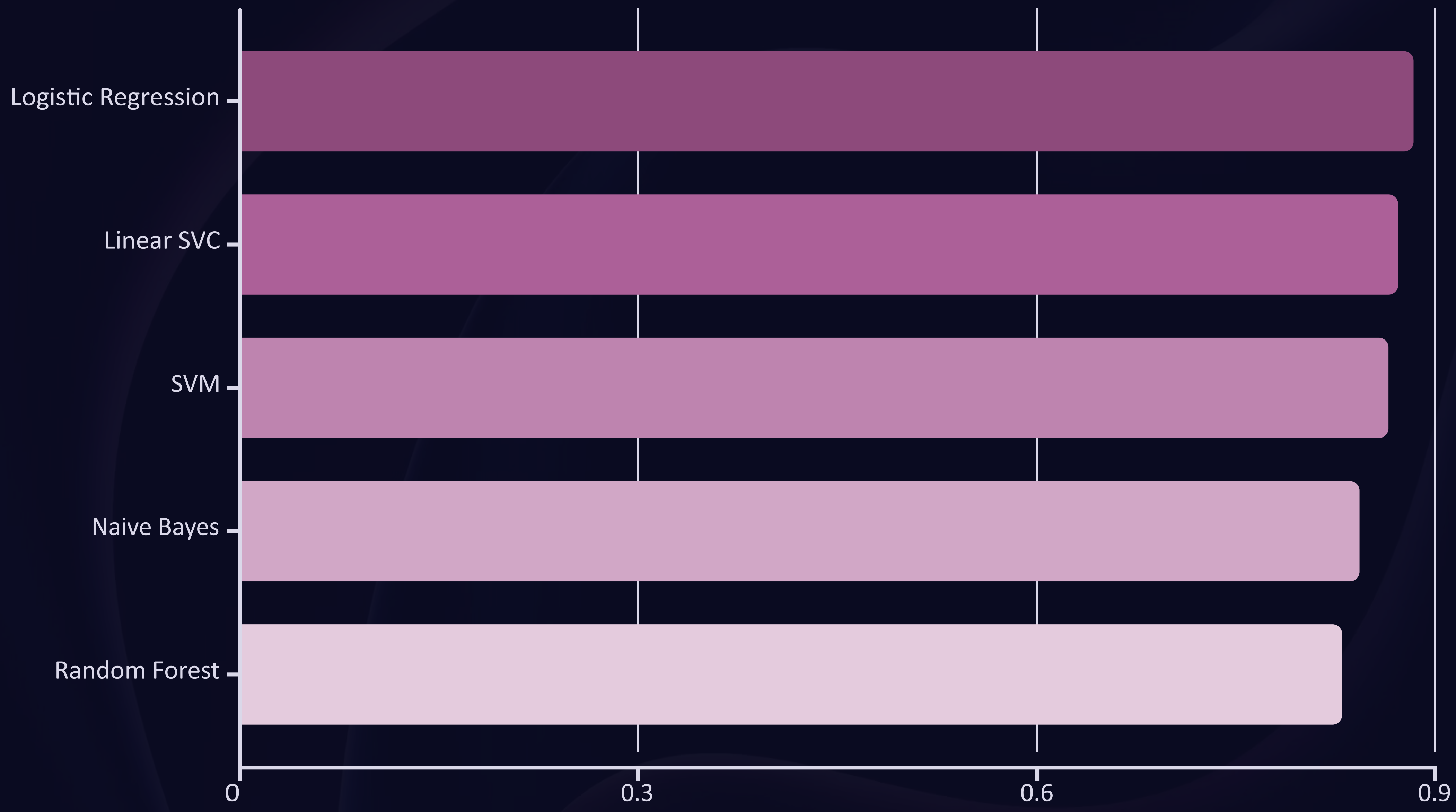
It's brilliant at identifying which words actually matter in a review. Common words like "the" get downweighted, whilst distinctive words like "masterpiece" or "disappointing" stand out.

The Power of 3-grams

Instead of just looking at single words, 3-grams capture short phrases like "not very good" or "absolutely loved it". This context is crucial for understanding sentiment 3 "not good" means something very different from "good"!

Model Selection: Finding the Winner

We didn't just pick a model and hope for the best. We tested five different classifiers to see which one would excel with our high-dimensional TF-IDF features.



Logistic Regression emerged as the clear winner, offering the best balance between performance and computational efficiency. It's fast to train and handles high-dimensional data brilliantly.

The Results:

88.4%

Overall Accuracy

Our model correctly classified sentiment in nearly 9 out of 10 reviews

~88%

F1-Score

Strong balance between precision and recall across both classes

50K

Reviews Analysed

Trained and tested on a substantial dataset for robust performance

These results validate our entire approach. The combination of careful pre-processing, TF-IDF 3-gram features, and Logistic Regression proved to be highly effective for large-scale sentiment classification.

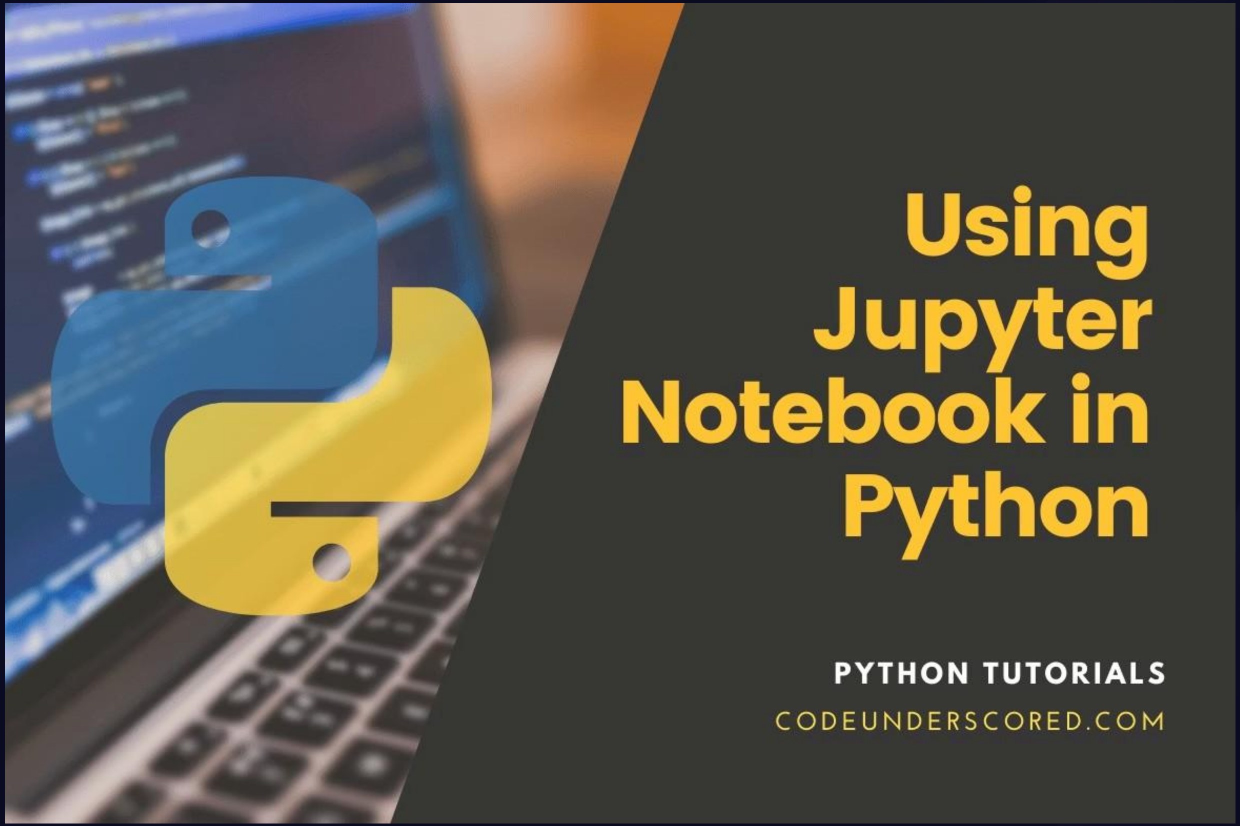
What's particularly impressive? The model maintained this accuracy on completely unseen test data, showing excellent generalisation ability.

Tech Stack & Implementation

Tools We Used

- **Python** 3 Our programming language of choice
- **Scikit-learn** 3 For ML models and evaluation
- **NLTK** 3 Natural language preprocessing
- **Pandas & NumPy**3 Data manipulation
- **Google Colab** 3 Development environment

The entire project is available in our Colab notebook, where you can see every step of the pipeline in action.



Key Takeaways

NLP pre-processing is absolutely critical

The quality of your features directly impacts model performance. Proper cleaning and feature extraction made all the difference.

Context matters in sentiment analysis

Using 3-grams instead of single words allowed our model to understand phrases and negations, significantly improving accuracy.

Sometimes simpler is better

Logistic Regression outperformed more complex models, proving that the right algorithm for your data beats complexity every time.

SENTI-MD demonstrates that with thoughtful design and solid fundamentals, you can build highly effective sentiment classifiers that deliver real-world value.